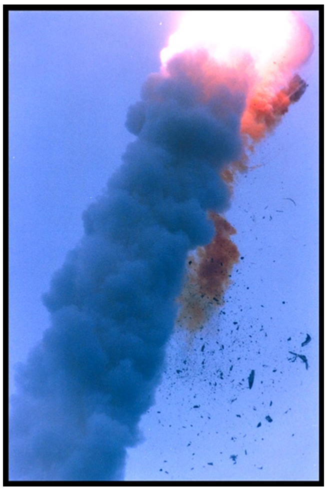
Introduction to Software Quality Assurance

From S. Somé, A. Williams

Introduction to Software Quality Assurance



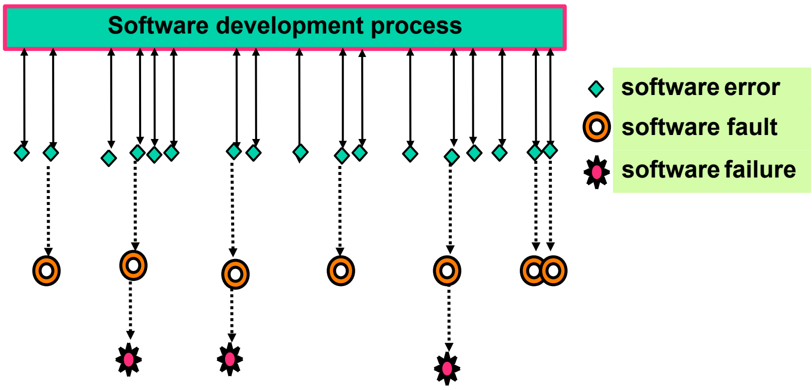
- Left: The Ariane 501 rocket's guidance computer software just threw an unchecked exception.
- [Image source: European Space Agency]

What is Software ?

- According to the IEEE
- Software is:
  - Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.

Software Errors, software faults and software failures

- Bug/defect/fault consequence of a human error.
  - Results in non-conformance to requirements.
  - Manifests as failure in running software.



Basic Definitions (Brugge and Dutoit)

- Reliability: The measure of success with which the observed behavior of a system confirms to some specification of its behavior.
- Failure: Any deviation of the observed behavior from the specified behavior.
- Error: The system is in a state such that further processing by the system will lead to a failure.
- Fault (Bug): The mechanical or algorithmic cause of an error.
  - ***Definitions vary across authors!***
- There are many different types of errors and different ways how we can deal with them.

Basic Definitions (Lethbride and Laganière)

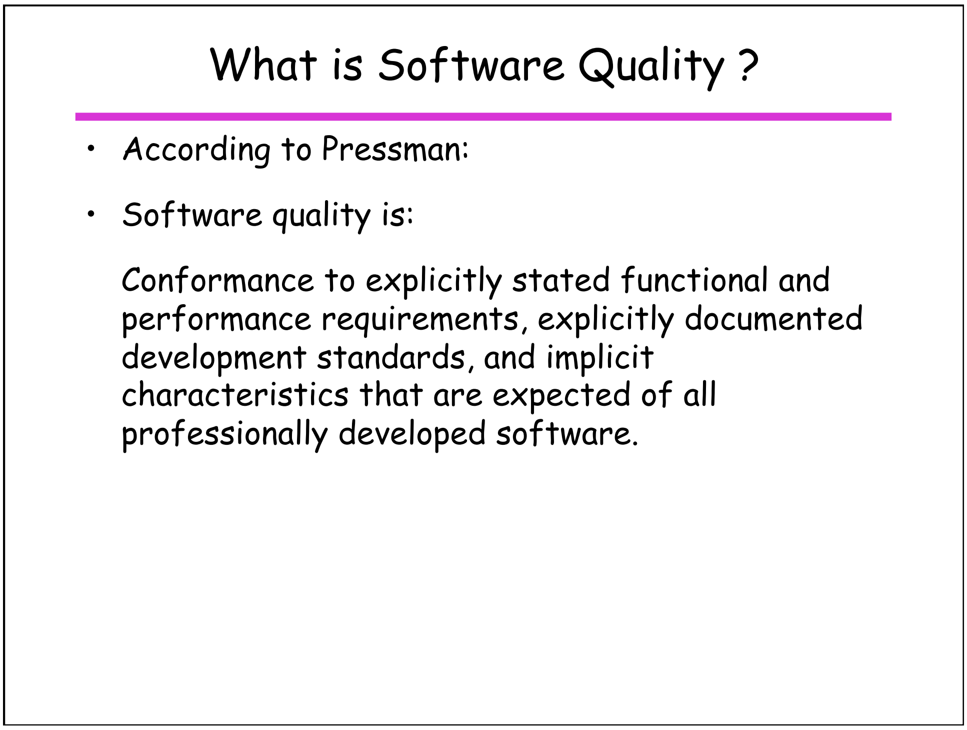
- A **failure** is an unacceptable behaviour exhibited by a system.
  - The frequency of failures measures the reliability.
  - An important design objective is to achieve a very low failure rate and hence high reliability.
  - A failure can result from a violation of an explicit or *implicit* requirement.
- A **defect** is a flaw in any aspect of the system that contributes, or may potentially contribute, to the occurrence of one or more failures.
  - It might take several defects to cause a particular failure.
- An **error** is a slip-up or inappropriate decision by a software developer that leads to the introduction of a defect.

Nine Causes of Software Errors

- Faulty requirements definition.
- Client-developer communication failures.
- Deliberate deviations from software requirements.
- Logical design errors.
- Coding errors.
- Non-compliance with documentation and coding instructions.
- Shortcomings of the testing process.
- User interface and procedure errors.
- Documentation errors.

What is Software Quality ?

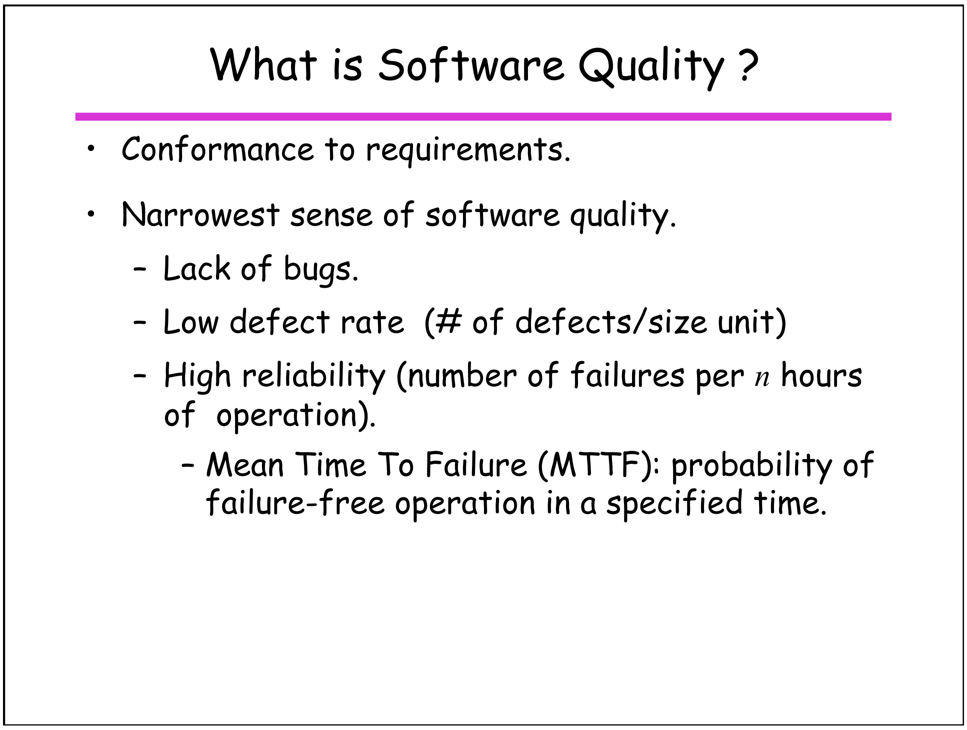
- According to the IEEE:
- Software quality is:
  1. The degree to which a system, component, or process meets specified requirements.
  2. The degree to which a system, component, or process meets customer or user needs or expectations.



What is Software Quality ?

- According to Pressman:
- Software quality is:

  Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.



What is Software Quality ?

- Conformance to requirements.
- Narrowest sense of software quality.
  - Lack of bugs.
  - Low defect rate (# of defects/size unit)
  - High reliability (number of failures per *n* hours of operation).
    - Mean Time To Failure (MTTF): probability of failure-free operation in a specified time.

When are defects introduced?

- The majority of defects are introduced in earlier phases.
  - Requirements (typically in Word…) are the top factor in a project's success or failure.

Phase	% of defects	% effort to fix
Requirements	56	82
Design	27	13
Code	7	1
Other	10	4

Cost of fixing defects

- Relative cost of fixing defects
  - benchmark: cost at requirements phase = 1

Phase found	Cost ratio
Requirements	1
Design	3 – 5
Coding	10
Unit / integration testing	15 - 40
System / acceptance testing	30 – 70
Production	40 - 1000

Importance of Software Quality

- Software is a major component of computer systems (about 80% of the cost) – used for
  - Communication (e.g. phone system, email system).
  - Health monitoring.
  - Transportation (e.g. automobile, aeronautics).
  - Economic exchanges (e.g. e-commerce).
  - Entertainment.
  - etc.
- Software defects are extremely costly in term of
  - money
  - reputation
  - loss of life

Importance of Software Quality

- Notable consequences of software errors:
  - 1988 shooting down of Airbus 320 by the USS Vincennes - cryptic and misleading output displayed by tracking software.
  - 1991 patriot missile failure - inaccurate calculation of time due to computer arithmetic errors.
  - London Ambulance Service Computer Aided Dispatch System – several deaths.
  - On June 3, 1980, the North American Aerospace Defense Command (NORAD) reported that the U.S. was under missile attack.
  - First operational launch attempt of the space shuttle, whose real-time operating software consists of about 500,000 lines of code, failed - synchronization problem among its flight-control computers.
  - 9 hour breakdown of AT&T's long-distance telephone network - caused by an untested code patch.

Importance of Software Quality

- Ariane 5 explosion June 4, 1996
  - First flight of the European Ariane 5 launcher crashed about 40 seconds after takeoff.
  - Cost was about half a billion dollars.
  - Explosion was the result of a software error:
    - Uncaught exception due to floating-point error: conversion from a 64-bit integer to a 16-bit signed integer applied to a larger than expected number.
    - Module was re-used without proper testing from Ariane 4 .
      - Error was not supposed to happen with Ariane 4.
      - No exception handler.

Importance of Software Quality

- Mars Climate Orbiter - September 23, 1999
  - Mars Climate Orbiter, disappeared as it began to orbit Mars.
  - Cost about $US 125-million.
  - Failure due to error in a transfer of information between a team in Colorado and a team in California.
    - One team used imperial units (e.g., inches, feet and pounds) while the other used metric units for a key spacecraft operation.

Importance of Software Quality

- Mars Polar Lander - December, 1999
  - Mars Polar Lander, disappeared during landing on Mars.
  - Failure most likely due to unexpected setting of a single data bit.
    - Defect not caught by testing.
    - Independent teams tested separate aspects.

Importance of Software Quality

- Internet viruses and worms
  - Blaster worm (US$ 525 million)
  - Sobig.F (US$ 500 million – 1 billion)
- Exploit well known software vulnerabilities
  - Software developers do not devote enough effort to applying lessons learned about the causes of vulnerabilities.
  - Same types of vulnerabilities continue to be seen in newer versions of products that were in earlier versions.
- Usability problems

Importance of Software Quality

- Monetary impact of poor software quality (Standish group - 1995)
- 175,000 software projects/year - Average Cost per project
  - Large companies - $US 2,322,000
  - Medium companies - $US 1,331,000
  - Small companies - $US 434,000
- 31.1% of projects canceled before completed.
  - cost $81 billion
  - Read “Death March Projects”

Importance of Software Quality

- 52.7% of projects exceed their budget - costing 189% of original estimates.
  - cost $59 billion
- 16.2% of software projects completed on-time and on-budget (9% for larger companies).
- Large companies - delivered systems have approximately only 42% of originally-proposed features and functions.
- 78.4% of smaller companies projects get deployed with at least 74.2% of their original features and functions.